Claims

1.   A programmable data processing device comprising :

- a loading engine (LE) for receiving portions of code of a first type and/or data from a stream (DC) of a broadcast network in which said portions are repeatedly transmitted,

    - a storage means (C) for storing the portions received by the loading engine,

    - an execution engine (EE) for executing an application embodied by the received portions,

    - a translating engine (TE) for translating the first type code into a native code of the execution engine (EE)

- characterised in that the translating engine (TE) is adapted to store the thus compiled portion in the storage means (C), to compile at least a certain one of said received portions into native code when a predetermined signalling information is received from a stream, and to interpret other portions of code, and that the execution engine (EE) is adapted to process compiled code and interpreted code within a same application.

2.   The data processing device according to claim 1, wherein the translating engine (TE) is adapted to select said certain portions according to control information received by it.

3.   The data processing device according to claim 1 or 2, wherein the stream (DC) is a DSM-CC carousel.

4.   The data processing device according to claim 3, wherein said portion is a DSM-CC module (I).

5.   The data processing device according to claim 3, wherein said portion is a fraction of a DSM-CC module.

6.   The data processing device according to one of claims 2 to 5, wherein the translating engine (TE) is adapted to receive said control information from said stream (DC).

7.   The data processing device according to claims 3, 4 and 6, wherein the control information is DSM-CC pre-fetch signalling.

8.   The data processing device according to claims 3 and 6 and one of claims 4 and 5, wherein the translating engine (TE) is adapted to extract the control information from a payload module (T) of the DSM-CC carousel (DC).

9.   The data processing device according to claim 8, wherein the translating engine (TE) is adapted to extract compiling optimization information relating to a portion of code to be compiled from said payload module (T) and to heed the compiling optimization in the process of compiling said portion of code.

10.   The data processing device according to one of claims 2 to 9, wherein the translating engine (TE) is adapted to receive control information from the execution engine (EE).

11.   The data processing device according to one of claims 1 to 10, wherein the translating engine (TE) is adapted to decide whether to compile or to interpret a given portion of first type code according to control information received from the execution engine (EE).

12.   The data processing device according to claim 11, wherein the translating engine (TE), during compilation of a given first type code portion, is adapted to ignore control information requiring said portion to be interpreted, and to finish compiling the portion.

13.   The data processing device according to claim 11, wherein the translating engine (TE) , when receiving control information requiring a given first type code portion to be interpreted during compilation of said portion, is adapted to abandon the compilation and to start interpreting the portion.

14.   A   data   processing   method,   comprising   the   steps   of:
        - a)   receiving (a1, a1') portions of code (I) of a first type and/ or data (D) from a stream (DC) of a broadcast network in which said portions (I,

D) are repeatedly transmitted, wherein the set of portions transmitted in said stream (DC) embodies one or more data processing applications

- b)   storing (a6, a9', b3) predetermined ones of said portions in a storage means (C),

- c)   when a predetermined signalling information is received from a stream, , compiling in a translation engine at least one of said portions comprising first type code into native code of an execution engine,

- d)   in the execution engine (EE), carrying out one of said data processing applications by executing (c6) the compiled native code (N) of the selected portions belonging to said one application and by interpreting (c7) non-selected portions of this application.

15.   The data processing method of claim 14, comprising, between steps c and d, the step of receiving an instruction from a user specifying the application to be carried out in step d.

16.   The data processing method according to claim 14 or 15, wherein in step c, said at least one portion is selected (a2, a3', a4', a5') based on control information supplied to the translation engine (TE).

17.   The data processing method according to one of claims 14 to 16, wherein the stream (DC) is a DSM-CC carousel.

18.   The data processing method according to claim 17, wherein said portion is a DSM-CC module.

19.   The data processing method according to claim 17, wherein said portion is a fraction of a DSM-CC module (I).

20.   The data processing method according to one of claims 16 to 19, wherein said control information is received (a2, a5') from said stream (DC).

21.   The data processing method according to claims 18 and 20, wherein the control information is DSM-CC pre-fetch information.

22. The data processing method according to one of claims 18 or 19 and claim 20, wherein the control information is a payload module (T) of the DSM-CC carousel (DC).

23. The data processing method of claim 22 wherein the control information further comprises compiling optimization information relating to a portion of code to be compiled, and the translation engine heeds the compiling optimization information when compiling said portion of code.

24. The data processing method according to one of claims 14 to 19, wherein said control information is received (a4') from said execution engine (EE).

25. The data processing method according to claim 24, wherein the translating engine (TE) decides based on said control information from the execution engine (EE) whether to compile or to interpret a given first type code portion.

26. The data processing method according to claim 25, wherein if the translation engine (TE) receives control information requiring a given portion to be interpreted during compilation of said portion, it ignores (c11) the control information and finishes compiling the portion.

27. The data processing method according to claim 25, wherein if the translation engine (TE) receives control information requiring a given portion to be interpreted during compilation of said portion, it abandons the compilation (c11') and starts interpreting the portion.

28. The data processing method according one of claims 14 to 27 in which, after step c), memory space allocated to the first type code of the compiled portion is released for overwriting.